VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

VYHLEDÁVÁNÍ PŘESNÝCH A PŘIBLIŽNÝCH REPETIC V SEKVENCI DNA NA ZÁKLADĚ POROVNÁVANÍ ZNAKŮ

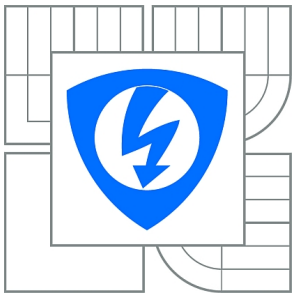
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PAVEL MARTAUS
AUTHOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF BIOMEDICAL ENGINEERING

VYHLEDÁVÁNÍ PŘESNÝCH A PŘIBLIŽNÝCH REPETIC V SEKVENCI DNA NA ZÁKLADĚ POROVNÁVANÍ ZNAKŮ

DETECTING EXACT AND APPROXIMATE REPEATS IN DNA BASED ON STRING MATCHING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

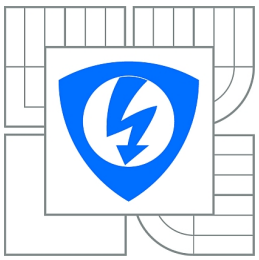
AUTOR PRÁCE — PAVEL MARTAUS

AUTHOR

VEDOUCÍ PRÁCE — Ing. VLADIMÍRA KUBICOVÁ

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Student:	Pavel Martaus	***ID:***	125053
Ročník:	3	***Akademický rok:***	2011/2012

NÁZEV TÉMATU:

Vyhledávání přesných a přibližných repetic v sekvenci DNA na základě porovnávaní znaků

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši o typech repetitivní DNA a algoritmech pro její vyhledávání z nukleotidových sekvencí. 2) Navrhněte algoritmus vyhledávání repetic v sekvenci na základě porovnávaní znaků využitím Hammingové vzdálenosti. Algoritmus navrhněte s ohledem na detekci nejenom identických repetic ale taktéž repetic, které mohou obsahovat mutaci – substituci. 3) Algoritmy realizujte v programovém prostředí Matlab a funkčnost ověřte na vybraných sekvencích. 4) Proveďte diskusi získaných výsledků a zhodnoťte účinnost a využitelnost řešení.

DOPORUČENÁ LITERATURA:

[1] KOLPAKOV, R. Finding approximate repetitions under Hamming distance, Theoretical Computer Science, vol. 303, pp. 135-156, 2001
[2] BENSON, G. Tandem repeats

Termín zadání:	6.2.2012	***Termín odevzdání:***	25.5.2012

Vedoucí práce: Ing. Vladimíra Kubicová

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

V této práci je v teoretické části popsána repetitivní DNA, jeji typy a metody pro jejich vyhledávání v sekvenci DNA. V praktické části je navržen a popsán algoritmus pro vyhledávání tandemových repetic s využitím Hammingové vzdálenosti a následně jeho zhodnocení a využití v budoucnosti.

KLÍČOVÁ SLOVA

Repetitivní DNA, analýza DNA sekvencí, Hammingova vzdálenost.

ABSTRACT

This work is described in the theoretical part of repetitive DNA, their types and methods to search in DNA sequence. In the practical part is designed and described an algorithm for finding tandem repeats using Hamming distance and consequently its evaluation and use in the future.

KEYWORDS

Repetitive DNA, analysis of DNA sequences, Hamming distance.

MARTAUS, Pavel *Vyhledávání přesných a přibližných repetic v sekvenci DNA na základě porovnávaní znaků*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2012. 35 s. Vedoucí práce byl Ing. Vladimíra Kubicová

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Vyhledávání přesných a přibližných repetic v sekvenci DNA na základě porovnávaní znaků“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno ....
(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucí bakalářské práce Ing. Vladimíře Kubicové za účinnou metodickou, pedagogickou a odbornou pomoc, konzultace a další cenné rady při zpracování mé bakalářské práce. Dále bych rád poděkoval své rodině a přítelkyni za trpělivost a podporu.

V Brně dne .....

.....
(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

DNA je tvořena dvěma částmi: kódující, která se přepisuje v proteiny a nekódující, která tvoří až 97% celé DNA. Některé její části mají regulační funkci, ovšem ve většině případů jejich funkce není známá. Pro vědce představují tyto prozatím ne zcela probádané části DNA, důležitou úlohu k vyhledávání podobnosti opakujících se sekvencí DNA. Pro jejich vyhledávání se používá různých metod jako například: vyhledávání repetetivních úseků ze spektra, pravděpodobnostního modelu (TRF) nebo s využitím Hammingovy vzdálenosti, na základě které je v následné druhé části vytvořený algoritmus, pro možnou detekci přesných tandemových repetic nebo detekci s mutací.

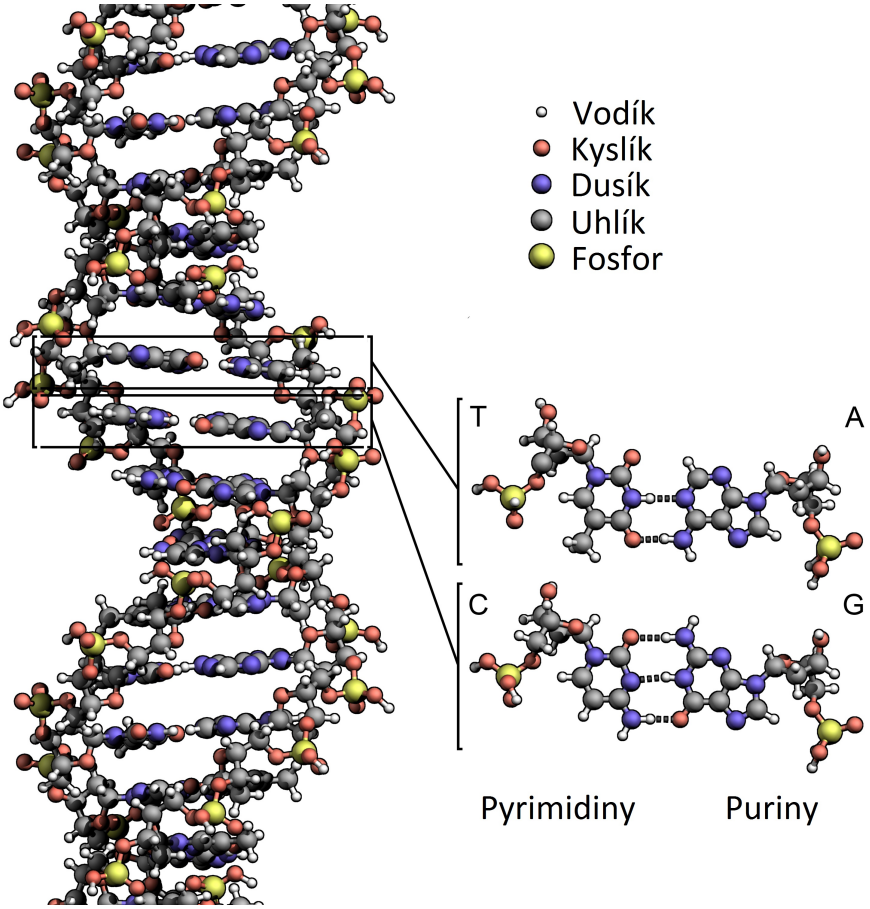
1 TEORETICKÝ ROZBOR

1.1 Deoxyribonukleová kyselina (DNA)

DNA je nukleová kyselina, která je nositelkou genetické informace všech organismů až na výjimky, jako jsou nebuněčné organismy, u nichž hraje úlohu RNA. DNA je biopolymer, který se skládá ze dvou řetězců nukleotidů a ty tvoří tzv. dvoušroubovici. Nukleotidy jsou vždy složeny z cukru deoxyribózy, fosfátové skupiny a jedné ze čtyř nukleových bází. Právě tyto báze v sobě obsahují informaci a mohou být buď purinové (adenin A, guanin G) nebo pyrimidinové (cytosin C, thymin T). Jednotlivé báze jsou spojeny vodíkovou vazbou a to následovně:

- $A \leftrightarrow T,\ T \leftrightarrow A$ (spojeny dvěma vodíkovými vazbami)
- $C \leftrightarrow G,\ G \leftrightarrow C$ (spojeny třemi vodíkovými vazbami)

DNA je tedy pro život nezbytnou látkou, která ve své struktuře kóduje a buňkám zadává jejich program a tím předurčuje vývoj a vlastnosti celého organismu. [2]

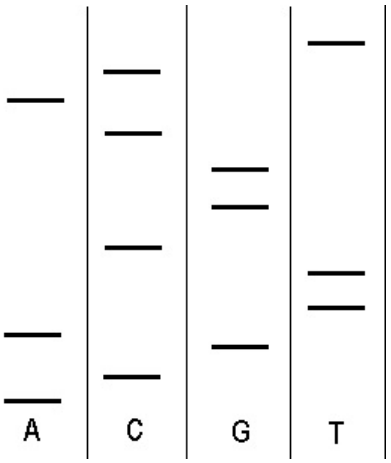


Obr. 1.1: DNA struktura. [6]

1.2 Sekvenování DNA

Sekvenování DNA je souhrnný termín pro metodu zjištění pořadí nukleových bází. Dnes je již známo několik metod sekvenování, ovšem nejznámější a nejpoužívanější je Sangerova metoda.

Tato metoda využívá ve své podstatě vlastnosti dideoxynukleotidů (ddATP, ddCTP, ddGTP a ddTTP), které nemají na 3' uhlíku ribosy OH skupinu. Tento fakt znamená, že na tento konec nemůže být navázán další nukleotid. Při použití elektrolýzy tím získáme, velké množství různě dlouhých oligonukleotidů, které budou všechny končit daným dideoxynukleotem např. při použití ddATP, bude oligonukleotid končit vždy adeninem. Pokud necháme proběhnout stejnou reakci, tentokrát s ddCTP, ddGTP a nakonec i ddTTP, dostaneme čtyři směsi oligonukleotidů, přičemž v každé směsi budou oligonukleotidy končit příslušnou bází. [7]



Obr. 1.2: Výsledek 4-dráhové elektroforézy

Čtení této sekvence probíhá od nukleotidu, který doputoval nejdál, tedy nejnižší hodnota (A) a dále se bude hledat nejbližší nukleotid, v tomto případě C. Tímto způsobem jsme schopni přečíst velmi malou sekvenci DNA.

V dnešní době se již využívá přístrojů tzv. sekvenátory, které značí fluorescenčními barvivy dideoxyribonukleotidy.

1.3 Repetitivní DNA

DNA eukaryot obsahuje značný podíl nekódujících sekvencí. Tak jako kódující DNA i nekódující může být unikátní, anebo se může nacházet v genomu ve více identických nebo podobných kopiích. Repetitivní DNA je tedy sekvence DNA s vysokým množstvím kopií. Pokud jsou kopie sekvenčního motivu v blocích a v řadě za sebou,

hovoříme o tandemových repeticích. Pokud jsou repetitivní sekvence rozptýleny v genomu, jedná se o rozptýlenou repetici. Funkce repetitivní DNA je stále nejasná, avšak v posledních letech bylo zjištěno, že počet tandemových repetic souvisí s nemocemi a hraje důležitou roli v regulaci genů. [1] [2]

1.3.1 Rozptýlená repetice

Rozptýlená repetice vzniká procesem transpozice ("skákání" segmentu DNA na jiné místo genomu). Rozlišují se dva typy transpozonů: transpozony a retrotranspozony.

Transpozony

Transpozony se dokážou přesouvat z místa na místo bez nutností replikace a vytvářet své kopie, čímž se genetická informace přenáší z jedné molekuly DNA (chromozómu) do jiné molekuly DNA. V lidském genomu jsou transpozony považovány za inaktivní, díky akumulaci mutací v průběhu fylogeneze obratlovců, tudíž můžeme najít pouze evoluční zbytky, tzv. "fosilie".

Retrotranspozony

Retrotranspozony jsou oproti transpozonům aktivní a tvoří až 45% lidského genomu. Retrotranspozony "expandují" mechanismem duplikace (copy and paste), a to díky RNA polymeráze (jako mnohé viry). Jsou přepsány do RNA a ta podléhá reverzní transkripci do DNA, která je vložena do genomu na nové místo. Z bezprostředního pohledu nemají retrotranspozony žádnou důležitou funkci v buňce - hovoří se o "starém harampádí" - odpadní DNA (junk DNA); nebo o sobecké DNA, neboť se transpozony propagují na úkor buněčných energetických zdrojů.

1.3.2 Tandemové repetice

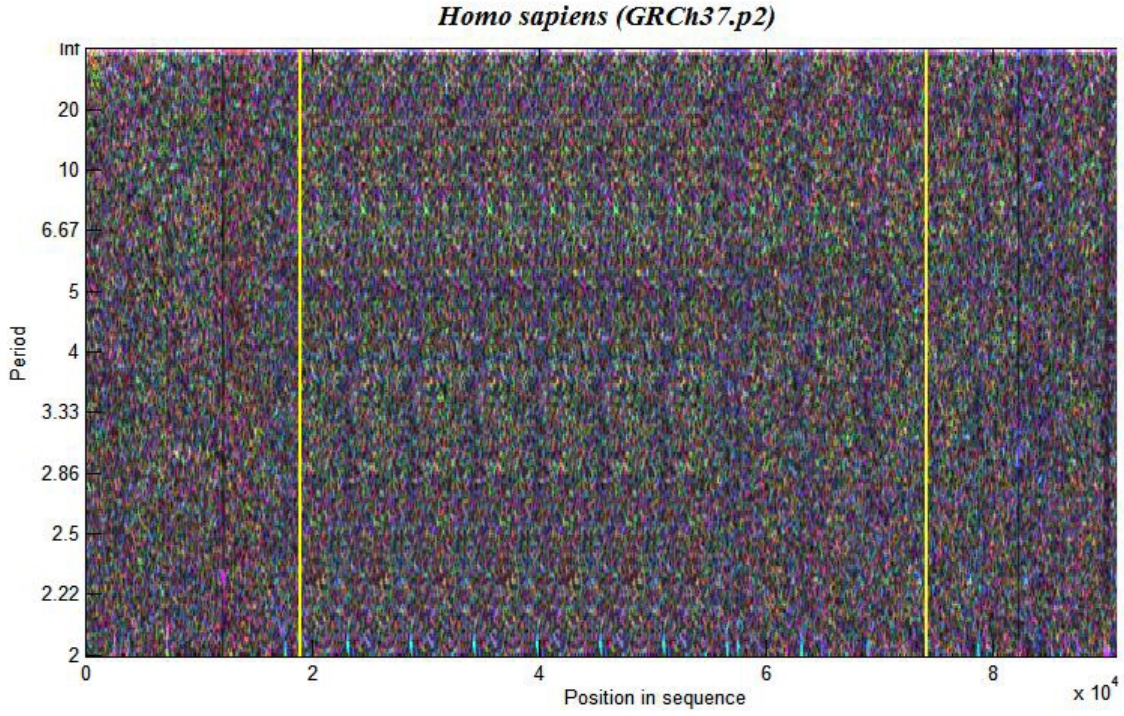
Tandemové repetice jsou tvořeny za sebou jdoucími téměř identickými až identickými jednotkami, ale jsou různé jak v délce repetice tak i celé repetice, tudíž se musí brát s odstupem.

Satelity

Satelity jsou největší repetice, které mají tendenci být složeny s dlouhých jednotek. Jméno dostali podle toho, že při rozbití DNA se při centrifugaci vytvoří ve zkumavce pruh v jiné výšce než zbytek chromozomu. Satelitní DNA je hojná v oblasti centromer a konstitutivního heterochromatinu. Opakující se vzor se pohybuje od 1bp až po blok několik Mbp. Je to úsek tvořený minisatelity a microsatelity.

Minisatelity

Minisatelity jsou kratší tandemové repetice, v rozsahu 1 až 20 kbp, které se více vyskytují v subtelomerických oblastech chromozomů. Někdy se uvažuje o tom, že by některé minisatelity mohly mít regulační funkce, jako např. VNTR v promotoru inzulinového genu, kde byla různá délka VNTR asociována s různými typy diabetu. Ve většině případů jsou vysoce polymorfní co do počtu opakování jednotky repetice a mohou být použity jako genetické markery - VNTR (variabilní množství tandemových repetic). Genetické markery jsou oblasti DNA, které mohou být jednoduše identifikovány a používají se při popisu variace druhů. V tomto případě se jedná o proměnný počet tandemových repetic na dané pozici. Na obr. 1.3 můžeme vidět ukázku tandemové repetice tvořené minisatelity. Jde vidět že mezi žlutými čárami je obraz pravidelný, kdežto mimo je rozmazaný.



Obr. 1.3: Tandemová repetice tvořená minisatelity.

Microsatelity

Mikrosatelity jsou zpravidla tvořeny opakováním 1-5 bp, o délce zřídka překračujícím stovky bp. Nejčastějšími jsou dinukleotidové repetice. Mikrosatelity jsou v genomu velice časté, vysoce polymorfní a jsou často používány jako genetické markery.

1.4 Vyhledávaní z nukleotidových sekvencí

Při vyhledávání nukleotidových sekvencí, se vyhledávají jak identické repetice, tak i repetice které jsou již pozměněny (mohou obsahovat mutaci - substituci).

Mutace

Mutace je změna genotypu oproti normálu. Mutace mohou nastat během replikace DNA, ovšem pravděpodobnost této chyby se pohybuje v řádech 10^{-7}. Tomuto se říká mutace spontánní. Většině mutací, tzv. indukovaných, je vyvoláno vnějšími mutagenními faktory (látky, které jsou schopny způsobit mutaci). Mezi mechanismy genových mutací jsou: **Adice** (vložení jednoho nebo více nadbytečných párů - prodloužení polypeptidového řetězce), **Delece** (ztráta jednoho nebo více nukleotidů - zkrácení polypeptidového řetězce) a **Substituce** (nahrazení báze původní sekvence za bázi jinou). Mutace jsou ve většině případů škodlivé, ale mohou být i užitečné a to z pohledu evoluce.

Při vyhledávání sekvencí DNA, využíváme převodu jednotlivých nukleotidů na čísla, což nám dává větší možnost použití technik zpracovávání signálů pro analýzu genomických dat. K jednotlivým nukleovým bázím se přiřadí buď celá čísla A = 1, G = 2, C = 3, T = 4, nebo komplexní čísla A = 1 + j, G = -1 + j, C = -1 - j, T = 1 - j nebo lze vyjádřit binárně. Na základě binárního pravidla mohou být DNA sekvence prezentovány jako

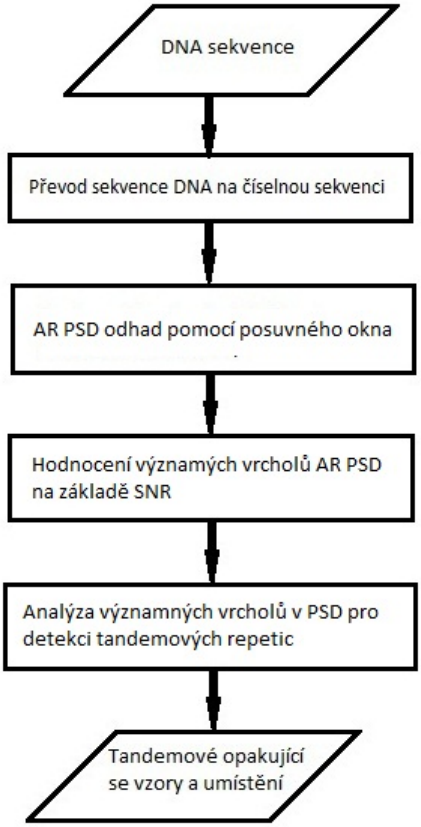
$$x[n] = xA[n] + xG[n] + xC[n] + xT[n], n = 0, 1, ..., N - 1$$

kde xA[n], xG[n], xC[n] a xT[n] jsou binární ukazatele a n je počet nukleotidů v sekvenci. Například pro sekvenci AGTCCGGTAAATGCCTTT se $xA[n]$ = 10000000111 0000000, kde 1 znamená přítomnost A a naopak 0 absenci A. Stejným způsobem lze odvodit xC[n], xG[n] a xT[n]. [3]

1.4.1 Vyhledávání repetitivních úseků ze spektra

Algoritmus pro analýzu je založen na spektrální analýze. Tato metoda nejen detekuje repetitivní část, ale odfiltruje i segmenty, které vypadají jako repetitivní ale nejsou. Proto je vhodná pro detekci přesných repetic. Tato metoda je založena na autoregresním (AR) modelu. Většina metod spektrální analýzy využívá Fourierovu transformaci, která je sice rychlejší, ale zkracují data a tudíž vytvářejí artefakty. Autoregresní model nabízí vysoké rozlišení, protože má schopnost extrapolace při vstupu autokorelační funkce. Postup pro detekci je následující:

Krok 1) Sekvenční převod: Převedení sekvence DNA na binární část.



Obr. 1.4: Vývojový diagram algoritmu pro detekci tandemových repetic. [3]

Krok 2) Spektrální odhad: Posuvné okno je aplikováno na čtyři binární sekvence a analyzuje jejich obsah poziční frekvence, kde je umístěné okno. Poté AR model v tomto okně postupně získá $P_A(\omega)$, $P_C(\omega)$, $P_G(\omega)$ a $P_T(\omega)$. PSD sekvence se získá součtem těchto jednotlivých spekter:

$$P_\omega = \sum_\alpha P_\alpha(\omega) \tag{1.1}$$

α ... A, T, G, C
ω ... úhlová prostorová frekvence

Pokud je DNA sekvence součástí času m, pak vrchol (peak) grafu lze vidět na úhlové frekvenci $\alpha = \frac{2\pi}{m}$ v PSD. Když budeme posouvat okno podél DNA sekvence, bude provádět lokalizovanou spektrální analýzu pozice a zobrazí výsledek pomocí spektrogramu.

Krok 3) Extrakt spektrálního vrcholu: Spektrogram zobrazí frekvenční polohu

roviny pro zobrazení repetic v sekvencích DNA. Nicméně, v některých případech se může obtížně určit, zda oblast obsahuje tandemové repetice pouhou vizuální kontrolou. Proto používáme SNR k posouzení významových vrcholů v každém posuvném oknu. SNR je definována jako:

$$R = \frac{P(\omega)}{\bar{P}} \tag{1.2}$$

kde průměr celkového spektra $\bar{P}$ se vypočte jako:

$$\bar{P} = \frac{1}{M} \sum_{n=0}^{M-1} P(\frac{n\pi}{M}) \tag{1.3}$$

$n\pi$ ... diskrétní úhlová frekvence

M ... vzorkovací rychlost

Z řady studií se zjistilo, že vrcholy R jsou významné tehdy, když jsou vyšší než 4 a proto se práh $H = R\bar{P}$ počítá pro každé okno. Posunováním okna vytvoříme graf prahu. Pokud maximální hodnota bude vyšší než práh, nastaví se 1, jinak je nastavena 0. Po odstranění všech hodnot pod hodnotu práhu, získáme graf s přibližným umístěním DNA.

Krok 4) Místo repetičního vzoru: Pomocí 3 parametrů $[(X_s, X_c)Y]$, kde $X_s a X_c$ je počáteční a koncový bod a Y je frekvence tohoto řádku, se vypočítá $A_f(i)$ a to tak, že se porovnává jednotlivé body s ostatníma s periodou T. Pokud se shoduje, připíše se 1. Pak se provede součet pomocí vzorce:

$$A_T(i) = \sum_{k=i}^{i+T-1} A_f(k) \tag{1.4}$$

Pomocí několika podmínek, dokážeme sestrojit tabulku jako je např.:

Gene	Region (bp)	Period	Repeat Unit	No. of Copies
X64775	49 – 57	3	TAC	3.0
	59 – 76	3	CGG	6.0
	141 – 188	3	GGC	15.7

Repeat sequence:
41 CGCACATGTA CTACGACACG GCGGCGGCGG CGGTGGACGA
81 GGCGCAGTTC TTGCGGCAGA TGGTGGCCGC GGCGGATCAC
121CACGCGGCCG CCGCTGGGAG AGGAGGCGGC GACGGCGACG
161GCGGCGGCGG CGGCGGCGGC GGTGGCGGGG AGAGGAAGCG

Obr. 1.5: Výsledek detekce tandemové repetice sekvence X64775.

V této tabulce lze rozpoznat, že se objevují mutace, ale podle daných kriterií je tato zanedbatelná mutace přijatelná. [3]

1.4.2 TRF - analýza na základě pravděpodobnostního modelu

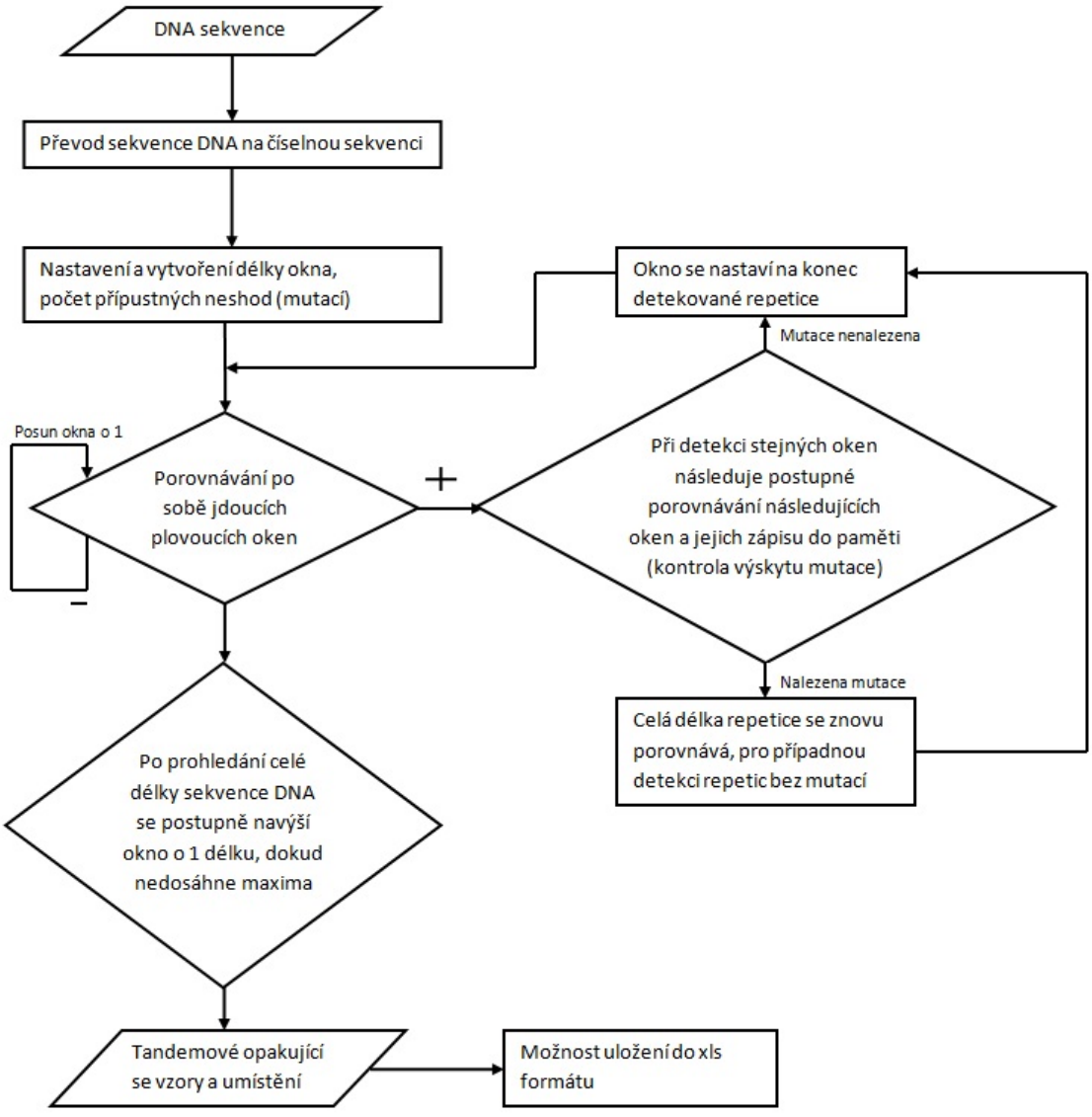
Je to algoritmus vytvořený k analýze tandemových repetic. Určuje procentuální shodu sousedních modelů tandemových repetic, frekvenci delecí, insercí a pomocí statistických kritérií pro uznání repetice jej lze využít. Dokáže detekovat tandemové repetice, které byly podrobeny rozsáhlé změně mutací pomocí analýzy čtyř úseků sekvencí: lidský gen frataxin, T-buněk a dvou kvasnicových chromosomů. [4]

1.4.3 Analýza na základě využití Hammingovy vzdálenosti

Tato metoda využívá Hammingovy vzdálenosti, což je počet pozic, při které se stejná délka sekvencí liší. Vyhledává nejen přesné repetice, ale i přibližné repetice obsahující mutaci, jelikož v biologii je veliký zájem pro jejich výzkum. [5] V následující části je detailně popsána celá metoda i s navrženým algoritmem.

2 NÁVRH ALGORITMU

Můj navržený algoritmus pracuje na základě Hammingovy vzdálenosti. Jak už bylo řečeno, je to počet pozic, při které se stejná délka sekvencí liší. Tento program je schopen vyhledávat jak repetice přesné, tak i repetice obsahující mutaci případně i velikost mutace. Jednotlivé sekce jsou rozděleny do několik podprogramů pro jeho přehlednost.



Obr. 2.1: Vývojový diagram programu na základě Hammingovy vzdálenosti.

2.1 Popis blokového diagramu

V následujících krocích bude popsán blokový diagram blok po bloku.

1.blok načte sekvenci DNA, kterou chceme zkoumat. Sekvence musí být ve formátu FASTA. Blok obsahuje i kontrolu, zda sekvence neobsahuje data jiná než nukleotidy a případně jej nahradí prázdným místem, které budou značit vždy mutaci.

2.blok zjistí délku sekvence a jednotlivým nukleotidům přiřadí číslici = převod na číslicovou sekvenci (A = 1, G = 2, C = 3, T = 4).

3.blok nastaví délku okna minimálně na 3 a maximálně na individuální hodnotu, kterou si nastavíme na vstupu. Pokud hodnota nebude nastavena, program se nespustí a vypíše chybu. Nastavíme hodnotu neshod, což je možný počet pozic, pro které se může báze(mutace) lišit. Tato hodnota by měla být ideálně nízká a při nulové hodnotě bude vyhledávat přesné repetice.

Ve **4.bloku** nastane nejdůležitější část programu a to, že okno se bude postupně posouvat po jedné délce a porovnávat shodu se sousedním oknem téže délky. Při nalezení shody, okno porovná i následující okna, pro možnou detekci déle opakující se repetitivní dna. Uloží hodnoty do paměti (perioda, místo a počet) a bude pokračovat v prohledávání, přičemž přeskočí nalezené místo obsahující tandemovou repetici. Pokud by nastala situace, že by nalezená repetice obsahovala mutaci, daná část sekvence se znovu zkontroluje, zda neobsahuje repetice bez mutace a vypíše se.

5.blok po prohledání celé sekvence, přesune okno na začátek sekvence a zvětší ho o jednu délku. Tím se zvětší délka hledajícího vzoru. Pátý blok se opakuje do té doby, než prohledá celou sekvenci při maximálním okně.

V **posledním bloku** program vypíše všechny opakující se tandemové repetice, jejich umístění, periodu a obsaženou mutaci. Případně si uživatel může uživatel výsledky uložit ve formátu xls.

2.2 Popis programu na příkladu

Tento příklad je ukázkový a bude vyhledávat přesné repetice. Vytvoříme si náhodnou sekvenci jako např.

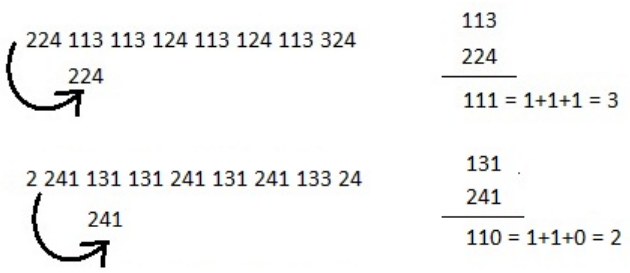
$$sekv.GGTAACAACAGTAACAGTAACCGT.$$

První blok načte tuto sekvenci.
Druhý blok zjistí délku sekvence $n = 24$ a převede ji na číselnou formu.

$$GGTAACAACAGTAACAGTAACCGT = 224113113124113124113324$$

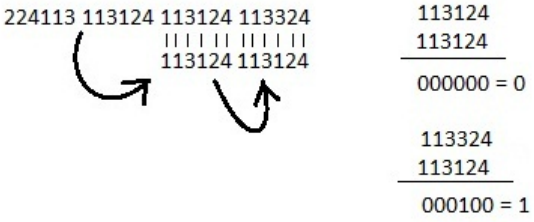
Třetí blok nastaví parametry, podle hodnot nastavených na vstupu. Pro náš případ je nastaven maximum okna = 8 a počet možných změn (mutace) = 0.

Čtvrtý blok začne prohledávat sousedící okna, jak můžeme vidět na následujícím modelu.



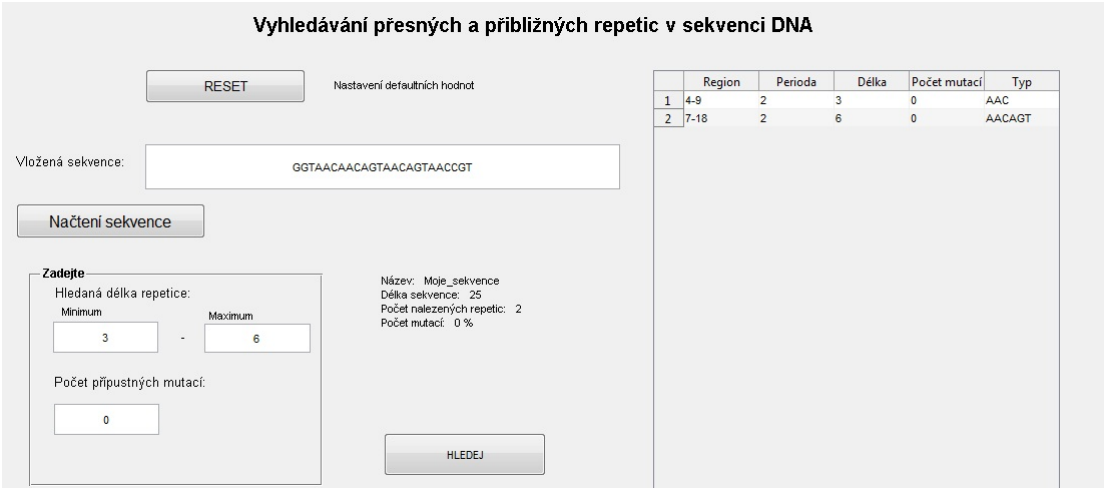
Okno o délce 3 se posouvá o jeden dílek a vypočítává Hammnigovu vzdálenost. Pro přesné repetice se musí Hammingova vzdálenost rovnat 0, což v tomto případě není.

Okno při detekci nulové Hammingovy vzdálenosti se posune o délku okna a porovná, jestli není další okno tandemová repetice. V tomto případě není, protože vzdálenost vyšla 2. Jelikož jsme zadali, že nevyhledáváme mutaci, uloží se repetice do paměti i s doprovodnými informacemi (region, délka, perioda, počet mutací).
Po prohledání celé sekvence, blok 5 navýší délku okna o 1.



V tomto případě byla nalezena stejná sousední sekvence o délce 6. Při porovnávání dalšího okna vyšla Hammingova vzdálenost 1, což naznačuje buď jinou sekvenci nebo sekvenci obsahující mutaci - substituci.

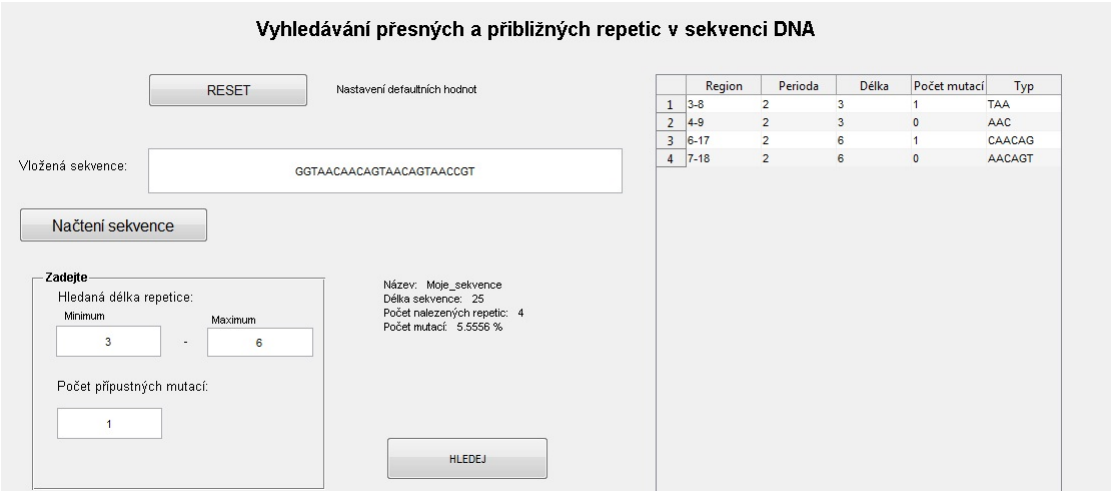
Po dosažení maxima okna se program ukončení a zobrazí se výsledek, neboli počet nalezených repetic, jejich délku, periodu, umístění a případně mutaci. Tento případ lze vidět na obr. 2.2.



Obr. 2.2: Výsledek vyhledávání bez mutací

Při zadání vstupní hodnoty neshod > 0, by byla vyhledána tandemová repetice obsahující mutaci. Tento program dokáže vyhledat mutaci obsahující substituci, jelikož je zachována stejná délka sekvence. Při deleci nebo inserci je změněna délka sekvence, a tudíž okno není schopné rozpoznat, zda chybí nebo naopak přebývá nukleotid.

Při nastavení možných mutací $= 1$, je výsledek na obr. 2.3.



Obr. 2.3: Výsledek vyhledávání s mutací

2.3 Ukázka zdrojového kódu

Program jsem napsal v programovém prostředí MATLAB, což je výkonný programovací jazyk pro vědecké a technické výpočty, zejména v maticových aplikacích. MATLAB byl implementován na všech významných platformách, jako jsou Windows, Linux a Mac. MATLAB obsahuje velké množství knihoven, které pokrývají prakticky všechny oblasti lidské činnosti a díky otevřené architektuře je uživateli umožněno vytvářet funkce dle své potřeby. Tyto knihovny jsou neustále vyvíjeny a rozšiřovány dle vývoje vědních a technických oborů. Pro můj algoritmus je tento program zcela vhodný, jelikož je přehledný a jeho logické operace jsou lehce pochopitelné. Program je rozdělen do dvou částí, kde hlavní program je z větší části uživatelské rozhraní GUI, což je uživatelské rozhraní sestavené z grafických objektu (komponent) jako jsou tlačítka, textová pole, posuvné seznamy, nabídky apod. Poskytuje rozhraní mezi uživatelem a aplikaci podřízeným kódem. V jednotlivých blocích hlavního programu jsou určité podmínky pro vyhledávání (např. max $>$ min, načtení sekvence, atd.).

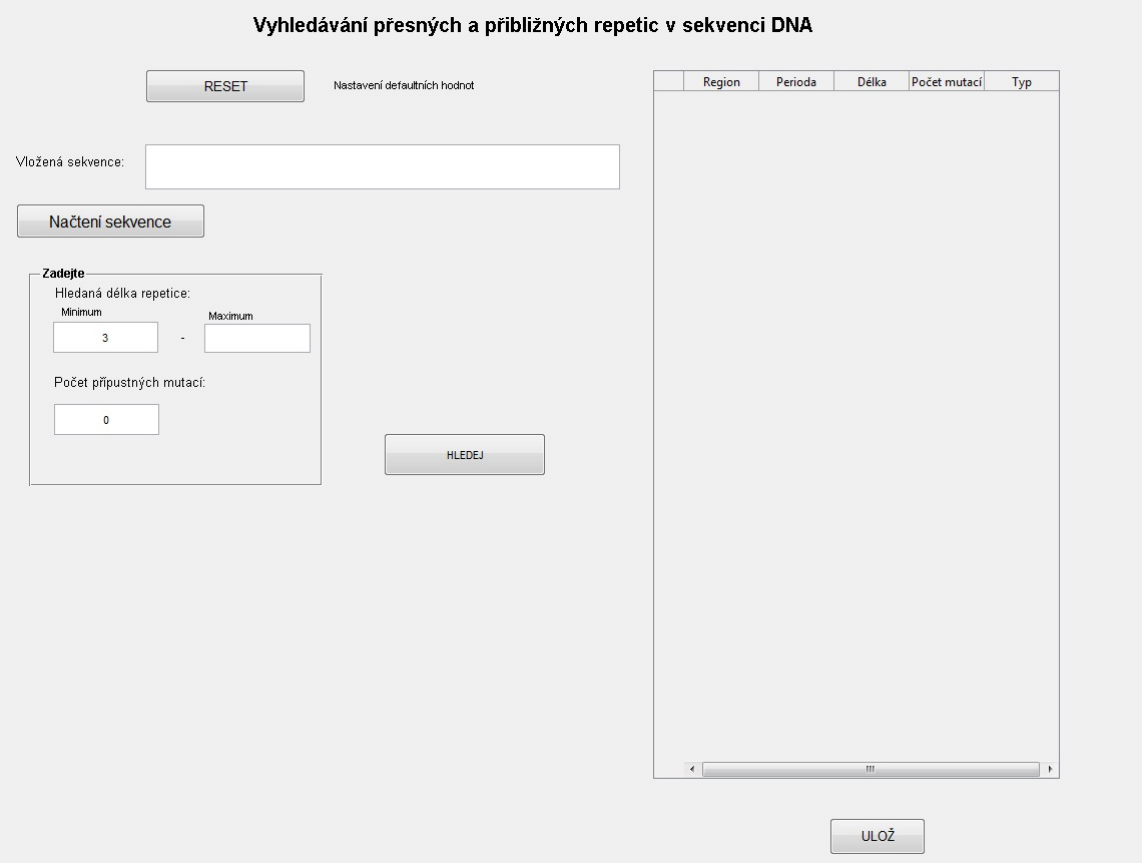
Druhá část programu slouží k převodu sekvence DNA na číselnou sekvenci a poté k jejímu porovnávání. Začíná od minimální délky okna a porovnává postupně po sobě jdoucí okna pomoci Hammingovy vzdálenosti a postupně se navyšuje.

Při spuštění programu se nám otevře grafické rozhraní GUI, kde vidíme několik oken. Tlačítko "RESET" slouží pro vymazání všech hodnot, včetně všech proměnných, výpisů tabulek a vrátí vše do původního stavu. Tlačítko "Načtení sekvence" nám otevře okno pro výběr souboru fasta, otevře jej a ve vedlejším okně se část sekvence ukáže. Současně se objeví informace o jejím názvu a délce sekvence. Pro nastavení délky okna a dovolenou mutaci, nám slouží malé okno pro vyplnění. Je zde vložena podmínka, že minimum nesmí být menší než 3 a maximum nesmí být menší než minimum. Tlačítko "HLEDEJ" nám prohledá danou sekvenci a ve vedlejší tabulce vypíše nalezené repetice, její délku, region, počet mutací a typ. V malé části se nám zobrazí i název, délka sekvence, počet nalezených repetic a počet mutací v procentech, který se vypočítá:

$$\bar{p} = \frac{S_p}{S_m} 100[\%] \tag{2.1}$$

S_p ... součet všech délek nalezené repetice
S_m ... součet všech výskytů mutací

Tlačítko "ULOŽ" uloží tabulku nalezených repetic do formátu xls, s počátečním jménem a umístěním kde se nachází program. V programovém prostředí GUI se objeví informace, že tabulka byla uložena.



Obr. 2.4: Uživatelské prostředí GUI

Nalézají se zde i prozatím neviditelná okna, která nám později poskytnou informaci o případné chybě a potřebný čas k výpočtu.

2.3.1 Načtení sekvence

Po stisknutí tlačítka pro načtení sekvence, se spustí část programu, kde se vyhledá cesta k fasta souboru, načte se, odstraní první řádky obsahující přebytečné informace a uloží se do poměti pouze sekvence a jméno souboru. Program není ošetřen pro načtení jiných souborů, než pro soubory typu fasta.

$[nazev, cesta] = uigetfile('*.*')$; % načtení daného souboru fasta
nazevcel = nazev;
n = length(nazev);
% zjištění délky sekvence nazev = nazev(1:n-6); % uloží do paměti jenom název (bez přípony fasta)
hlavicka = [];

sekvence = [];
fid = fopen(nazevcel); % otevře soubor
tline = fgetl(fid);
hlavicka = tline;

while ischar(tline) % uloží pouze sekvenci, bez prvotních informací
tline = fgetl(fid);
sekvence = [sekvence tline];
end

fclose(fid);

2.3.2 Podmínky

Program obsahuje několik podmínek pro spuštění vyhledávače. Po stisku tlačítka HLEDEJ, se provede kontrola vstupních údajů a korekce maxima a minima. Při nalezení chyby program vyhledá o kterou chybu jde, v opačném případě se spustí podprogram pro vyhledávání repetic. Pokud program nenalezne žádné repetice, vypíše do kolonky chyb, že daná sekvence neobsahuje žádnou repetice s uvedených podmínek.

$if sekvence(1) =' Q'$ && $max >= min$ && $max = 0$ && $min >= 3$ % vyskyt sekvence, podmínky hranic
set(handles.tabulka, 'data','');
set(handles.buck,'String','');
set(handles.cas, 'String','');
$[konec, casovac, informace] = prohledavani(min, max, mut, sekvence, nazev);$ % volání podprogramu
$if chyba = 0$
set(handles.tabulka, 'data', '');
set(handles.buck,'String',chyba);
if casovac > 60 % počítání více jak minuta
minut = floor(casovac/60);
secund = casovac-60*minut;
else % méně jak minuta
minut = 0;
secund = casovac;
end

```
time =['Čas potřebný k výpočtu: ',num2str(minut),' min. ',num2str(secund),' sekund'];
set(handles.cas, 'String', time); % výpis potřebného času k výpočtu
else
set(handles.buck,'String',''); % vymazání chyb
set(handles.tabulka, 'data', konec) % výpis dat
set(handles.info, 'String', informace) % výpis informací
if casovac > 60 % počítání více jak minuta
minut = floor(casovac/60);
secund = casovac-60*minut;
else % méně jak minuta
minut = 0;
secund = casovac;
end
time =['Čas potřebný k výpočtu: ',num2str(minut),' min. ',num2str(secund),' sekund'];
set(handles.cas, 'String', time) % výpis potřebného času k výpočtu
elseif sekvence(1) == 'Q' % nalezení přesné chyby
set(handles.buck,'String','CHYBA: Není načtena sekvence');
elseif max == 0
set(handles.buck,'String','CHYBA: Maximum nenalezeno');
elseif max < min
set(handles.buck,'String','CHYBA: Maximum musí být větší jak minimum');
elseif min < 3
set(handles.buck,'String','CHYBA: Minimum musí být větší nebo rovno hodnotě
3');
else
set(handles.buck,'String','CHYBA: Neznámá chyba');
end
```

2.3.3 Prohledávání sekvence

Při spuštění podprogramu se spustí časovač, který se zastaví až po ukončení vyhledávání (zjištění závislosti vyhledávání na velikosti sekvence, délky okna a počtu mutací). Program prvně převede sekvenci DNA na číselnou sekvenci,

```
n = length(sekvence); % zjištění délky sekvence
sek = []; % nastaneví prázdných hodnot pro pozdější výpočet
for i = 1:n % Přeměna znaků na čísla
switch sekvence(i)
```

```
case 'A','a'
sek(i) = '1';
case 'C','c'
sek(i) = '3';
case 'G','g'
sek(i) = '2';
case 'T','t'
sek(i) = '4';
otherwise
sek(i) = ' ';
end
end
```

poté se spustí do hlavního vyhledávání. Část programu pro vyhledávání se při mutaci opakuje 2krát s menší změnou (druhá část programu vyhledává jenom přesné repetice a je spuštěn jen při nalezení repetice s mutací), pro případné zjištění repetice bez mutace.

```
for i = min : max % prohledávání od minima po maximum
odp = n - ((2 * i) - 1); % výpočet posledního místa pro danou dálku sekvence při maximální délce
no3=0;
for j = 1:odp % prohledávání celé délky sekvence
if j > no3 % při nalezení repetice se již daná sekvence nebude prohledávat
zac = sek(j:j+i-1); % vytýčení začátku
kon = sek(j+i:i+j+i-1); % vytýčení konce
pocet = 0; % kontrola nelezení stejné repetice 0 = true , 1 a více = false(mutace)
opak = 1; % počet nalezených opakujících se repetic
citac = 0; % počítání kroků
while pocet <= mut % opakování pokud je nalezená shoda
for k = 1 : i % porovnávání vedlejších částí sekvence
if zac(k)==kon(k)
else pocet = pocet+1; % pocet neshodujících se sekvencí
end
end
citac = citac + 1; % počet kroků
if pocet <= mut
pocet_kon = pocet;
opak = opak+1;
```

```
zac1 = (citac*i)+j;
zac2 = (citac*i)+(j+i-1);
kon1 = (citac*i)+(j+i);
kon2 = (citac*i)+(i+j+i-1);
end
if pocet <= mut&&kon2 <= n
kon = sek(kon1 : kon2);
elseif opak >= 2|||(pocet <= mut&&kon2 > n)
celk = celk+1; % čítač uložených sekvencí

seznam(celk) = struct('Typ', sekvence(j : j + i - 1),' Oblast_zac', j,' Oblast_kon',
zac2,' Pocet_opakovani', opak,' Delka_vzoru', i,' Pocet_mutaci', pocet_kon);
pocet = 1000; % kontrola že už se nebude dál načítat opkaující se sekvenci na konci
sekvence
no3 = zac2;

end
end
end
end
```

2.4 Ukázky výsledků

Svůj program jsem vyzkoušel na několika sekvencích. Z internetových stránek *http* : *//www.ncbi.nlm.nih.gov/* jsem stáhl část sekvence určitých zvířat: Rattus norvegicus.fasta, Homo sapiens.fasta, Macaca mulatta.fasta, Musca domestica.fasta, Vulpes vulpes.fasta, ad. Porovnával jsem počet vyskytujících se repetic, ale i čas potřebný k výpočtu v závislosti na délce sekvence. V následujících obrázcích lze vidět rozdíl mezi vyhledávání přesných repetic (obr. 2.7) a vyhledávání repetic přesných tak i s mutací (obr. 2.8). U těchto výsledků lze vidět, že při vyhledávání repetic přesných i repetic s mutací program počítal přibližně 3krát oproti prvnímu (obr. 2.7) déle a našel přibližně 3krát více repetic. Dále lze vidět, že u obr. 2.8 byla uložena data do formátu xls. Při zkoumání nalezených repetic jsem usoudil, že nejčastěji vyskytující se repetice které jsem zkoumal mají délku 3 a počet period 2 s občasným výskytem 3.

Pro zjištění závislosti délky výpočtu, jsem vytvořil tabulky pro několik sekvencí a ty následně vynesl do grafu. Z prvního grafu jasně vyplývá, že ze zvětšující se délky

repetice, se nám lineárně zvyšuje doba vyhledávání. U druhé tabulky jsem si vytvořil dvě sekvence. Obě dvě sekvence byly stejně dlouhé, přičemž první obsahovala samé repetice, druhá neobsahovala žádné. Z druhého grafu jsem došel k názoru, že při vyhledávání přesných repetic je rychlejší sekvence neobsahující repetice, kdežto při vyhledávání s mutací je vyhledávání sekvence neobsahující repetice až 9x delší.

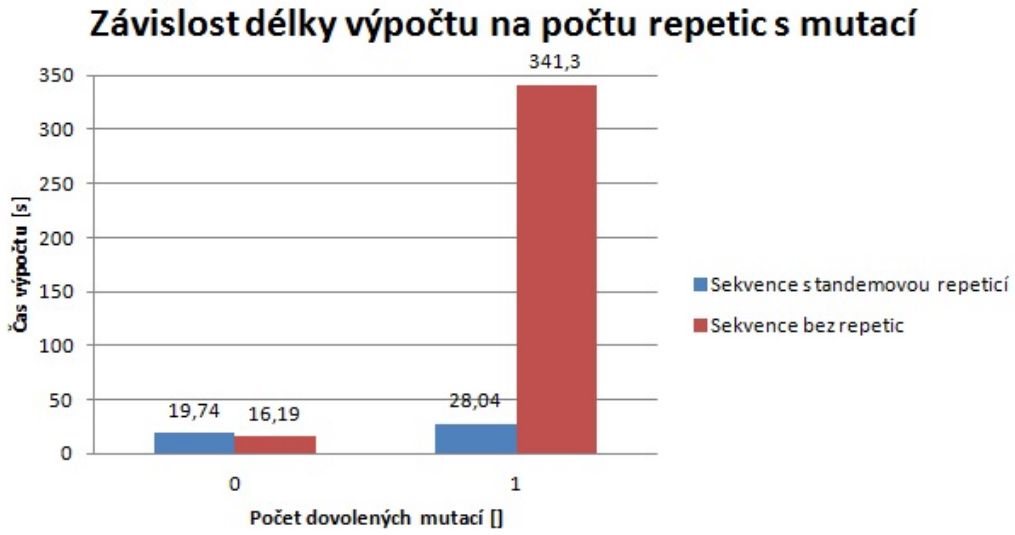
Název zvířete	**čas[s]**	**Délka sekvence**
Canis_lupus	0,96	1423
Homo_sapiens	1,32	1911
Delphinus_delphis	0,68	1045
Felis_silvestris	1,86	2717
Macaca_mulatta	5,01	6856
Mus_musculus	4,59	6239
Musca_domestica	2,41	3445
Panthera_leo	1,01	1513
Rattus_norvegicus	1,60	2359
Vulpes_vulpes	0,67	1039

Tab. 2.1: Závislost délky výpočtu na délce sekvence

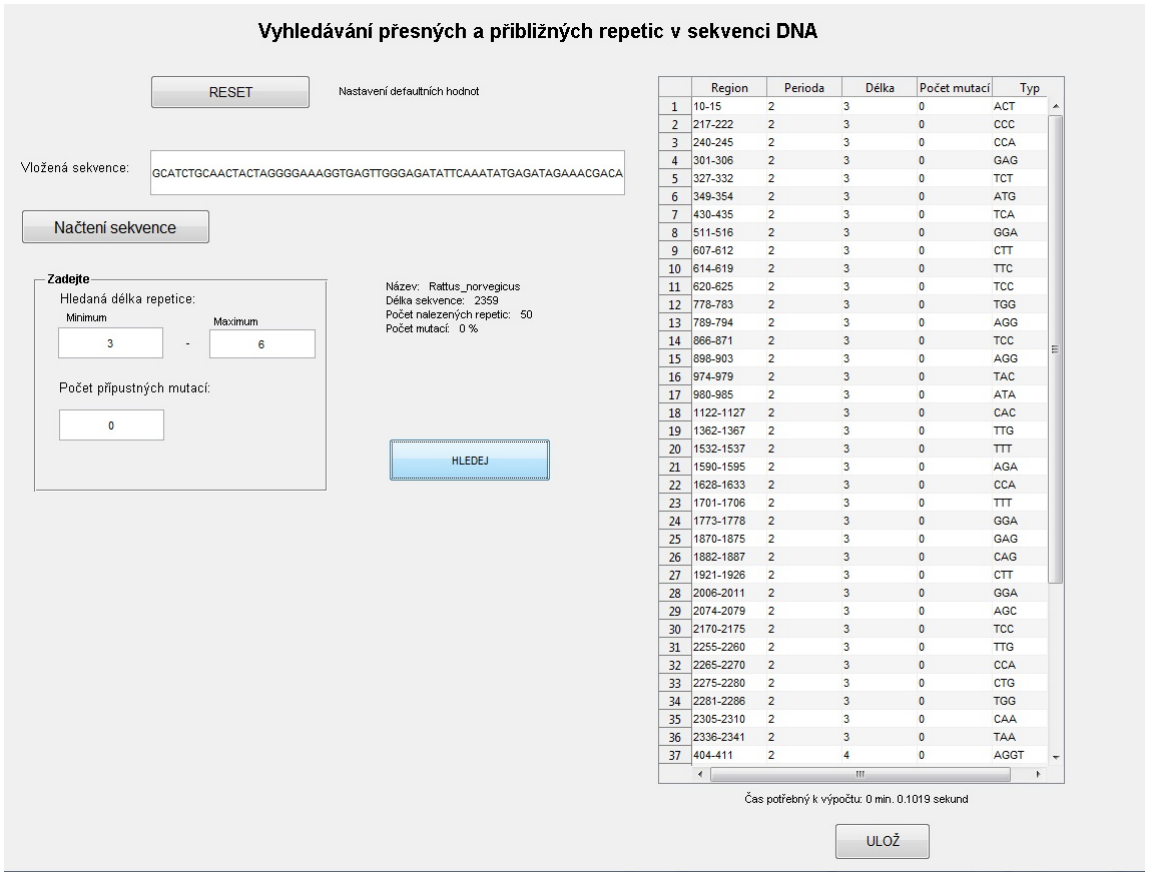
Název sekvence	**čas[s]**	**Počet repetic**	**Dovolené mutace**
Sekvence s tandemovou repeticí	19,74	5318	0
Sekvence s tandemovou repeticí	28,04	13293	1
Sekvence bez repetic	16,19	0	0
Sekvence bez repetic	343,3	4524	1

Tab. 2.2: Závislost délky výpočtu na počtu repetic s mutací

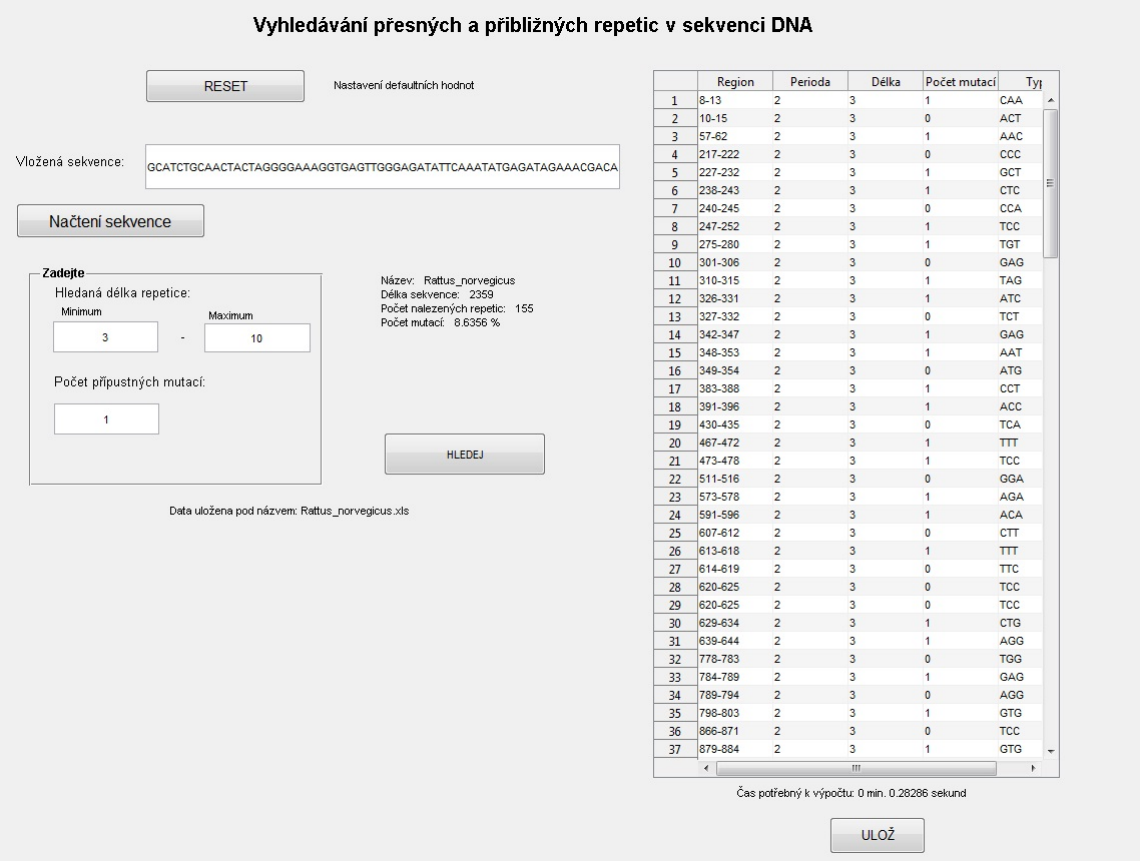
Obr. 2.5: Závislost délky výpočtu na délce sekvence



Obr. 2.6: Závislost délky výpočtu na počtu repetic s mutací



Obr. 2.7: Výsledek pro vyhledávání přesných repetic



Obr. 2.8: Výsledek pro vyhledávání přesných repetic a repetic s mutací

2.5 Zhodnocení

Výsledky programu jsem měl možnost porovnat s některými, již nalezenými repeticemi z jiných programů. V literatuře Hongxia, Zhou [3] je pár výsledků, které jsou uvedeny. Např. z obr. 1.5, kde vyhledávali repetice ze sekvence X64775, jsem dosáhl ne stejných, ale velmi podobných výsledků. Tento rozdíl mohl nastat jak v algoritmu, tak v postupu prohledávání s ohledem na výskyt mutace.

Program zvládl i zátěžový test, kdy prohledával v kuse 6 hodin sekvenci o délce přes milion nukleotidů, tudíž zvládá i obsáhlé sekvence.

Tento program určitě není konečný a dá se i nadále upravovat a vylepšovat, jak v úspěšnosti hledání, tak i v estetické části. Program by mohl být doplněn o vyhledávání insercí nebo delecí. Při delším porovnávání (více než minutu), by se hodil přibližný odhad času výpočtu. V tomto bodě nastal problém, když program porovnával data v kuse několik minut, že se člověk stal nejistým, zda nenastala vnitřní chyba či nekonečná smyčka, a proto by bylo mnohem lepší, kdyby zde byl jakýkoliv procentuální ukazatel postupu. Může být doplněn i o seřazení podle typu, které chceme (max repetic, délka repetic) nebo délky, které chceme. Chybí načítání z jiných typů souborů, kdy program by mohl být schopen rozpoznat a načíst ze souboru jenom sekvenci, nebo položku pro ruční vložení (napsání) sekvence, kterou chceme prohledat atd.

3 ZÁVĚR

V této práci jsem se zabýval typy repetitivní DNA a jejich vyhledávání v sekvenci DNA. Vytvořil jsem algoritmus pro detekci repetitivní DNA na základě Hammingovy vzdálenosti. Potýkal jsem se s mnoha problémy při programování vyhledávání, od některých myšlenek jsem se musel stáhnout, jelikož byli pro mě zatím nerealizovatelné (vyhledávání ze souboru jiných než fasta). Naopak několik věcí jsem přidal navíc mimo původní plán. Největší problém jsem měl u algoritmu při vyhledávání repetice s mutací a její následné prohledání, zda neobsahuje přesné repetice. Program podle mého názoru je co se týče vyhledávání přesný. Dokáže vyhledat jak přesné repetice, tak repetice obsahující mutaci. V budoucnu by bylo vhodné ještě přidat vyhledávání repetic obsahující inserci nebo naopak deleci.

LITERATURA

[1] ŠEDA, O.: *Genetické haraburdí-repetitivní DNA*. Ústav biologie a lékařské genetiky 1. LF UK a VFN, [on-line], [cit. 2011-17-12], dostupné na internetu: <http://biol.lf1.cuni.cz/ucebnice/repetitivni_dna.htm>

[2] HONZÍKOVÁ, N.: *Biologie člověka*. Skriptum VUT FEKT, Brno 2003.

[3] Hongxia, Zhou et al.: *Detection of Tandem Repeats in DNA Sequences Based on Parametric Spectral Estimation*. IEEE transaction on information technology in biomedicine, Vol. 13, No. 5, 2009, s. 747-755.

[4] Benson, G.: *Tandem repeats finder: a program to analyze DNA sequences*. Oxford University Press, , Vol. 27, No. 2, 1999, s. 573-580.

[5] Kolpakov, R., Kucherov, G.: *Finding Approximate Repetitions under Hamming Distance*. Institut National de Recherche en Informatique et en Automatique, 2001.

[6] *WIKIPEDIE*. [on-line], [cit. 2011-20-12], dostupné na internetu: <http://cs.wikipedia.org/wiki/Soubor:DNA_Structure%2BKey%2BLabelled.pn_NoBB_cs.png>

[7] Šípek, A.: *Genetika - Biologie*. [on-line], [cit. 2011-17-12], dostupné na internetu: <http://www.genetika-biologie.cz/mapovani-genomu>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A Adenin

AR Autoregresní model

bp Base pairs (Párů bazí)

C Cytosin

DNA Deoxyribonukleová kyselina

G Guanin

PSD Výkonové spektrum hustoty

RNA Ribonukleová kyselina

T Thymin

VNTR Variable number of tandem repeats (Variabilní množství tandemových repetic)

4 OBSAH PŘILOŽENÉHO CD

1. Textová část práce ve formátu *.pdf
2. Vytvořený program *DNAfind* v prostředí MATLAB
3. Vytvořený podprogram *prohledavani* v prostředí MATLAB
4. Sekvence Rattus_norvegicus, Canis_lupus, Homo_sapiens
5. Uměle vytvořené sekvence Moje_sekvence_repetice, Moje_sekvence_bezrepetice
6. Vědecké články citované v práci